IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of: Jeffrey G. Cheng et al.	Examiner: Philip A. Guyton
Application No.: 10/672,180	Group Art Unit: 2113
Filed: September 26, 2003	Docket No.: 00100.03.0032
For: **METHOD AND APPARATUS FOR MONITORING AND RESETTING A CO-PROCESSOR**	

RESPONSE TO NOTIFICATION OF NON-COMPLIANT APPEAL BRIEF UNDER 37 C.F.R. § 41.37

Dear Sir:

Appellant submits this revised Argument Section further to the Notification of Non-Compliant Appeal Brief Under 37 C.F.R. § 41.37 dated May 6, 2009, in the above-identified application. A heading for the rejection of claims 12 and 13 has been added on page 12.

Appellant respectfully believes that the Appeal Brief is now compliant. If there are any questions, the Patent Office is invited to contact the below listed attorney.

Respectfully submitted,

Date: June 8, 2009

By: /Christopher J. Reckamp/
Christopher J. Reckamp
Registration No. 34,414

Vedder Price P.C.
222 N. LaSalle Street
Chicago, Illinois 60601
PHONE: (312) 609-7599
FAX: (312) 609-5005

VII. ARGUMENT: FORSMAN FAILS TEACH EACH AND EVERY LIMITATION OF CLAIMS 1-2, 5-7, 10-11, 24-26, 28-30 AND 32-35

To establish a *prima facie* case of anticipation, each and every limitation of a claim must be found, either expressly or inherently, in a single prior art reference. MPEP § 2131. Because the Forsman fails to teach each and every limitation of claims 1-2, 5-7, 10-13, 24-26, 28-30 and 32-35[1], the rejections to these claims should be reversed and the claims should be allowed.

A. CLAIMS 2, 6-7, 12-13 AND 24 STAND OR FALL TOGETHER WITH INDEPENDENT CLAIM 1

Claims 1-2, 6-7 and 24 stand rejected under 35 U.S.C. § 102(e) as allegedly being anticipated by Forsman. Claims 12 and 13 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Forsman in view of Hill. Claims 2, 6-7, 12-13 and 24 stand or fall together with claim 1. Claim 1 requires a circuit for monitoring and resetting a co-processor comprising:

> a hang detector module operative to detect a hang in the co-processor <u>by detecting a discrepancy between a current state of the co-processor and a current activity of the co-processor</u>; and
>
> a selective processor reset module operative to selectively reset the co-processor without resetting a processor, in response to detecting a hang in the co-processor. (Emphasis added).

The final Office action mailed October 30, 2007 (the "Final Office Action") alleges that Forsman discloses all aspects of Appellant's claim 1. (Final Office Action, p. 2.) Appellant disagrees because Forsman is limited to a system and method that (1) resets a service processor when a host processor fails to detect a heartbeat signal from the service processor; or (2) resets a service processor when (a) a host processor fails to detect a heartbeat signal from the service

[1] Claims 12 and 13 are rejected under 35 U.S.C. § 103(a). However, because Forsman does not teach what is alleged, the obviousness rejections are also improper and should be reversed.

processor and (b) the host processor determines that conditions do not exist that preempt the host processor from resetting the service processor. In the first instance, Forsman fails to detect any sort of discrepancy between two things such as a current state and a current activity of a co-processor. In the second instance, Forsman resets the service processor only after determining a consistent condition between the lack of heartbeat signals and a lack of preempting conditions.

(i) *The Forsman Reference*

Forsman appears to be directed to a system and method for resetting a service processor and reestablishing communications between a host processor and a service processor after the service processor has ceased to function correctly. (Forsman, Title & Abstract.) Forsman teaches that "[i]n a proper running state … [h]ost 202 ("host processor") and service processor 204 … exchange heartbeat signals 206." (*Id.*, Col. 4, ll. 9-12.) In a first embodiment, if the host processor 202 fails to detect a heartbeat signal 206 from the service processor 204, the service processor 204 is not functioning correctly and must be reset. (*Id.*, Col. 4, ll. 13-15, 25-35.) Specifically, a hard reset of the service processor 203 is initiated in a way that does not disturb host processor 202 usage of shared resources. *Id.*

In a second embodiment where the host processor 202 initiates the communication recovery or hard reset of the service processor 204, host processor 202, after detecting alack of heartbeat signals, "checks the status portion of the status/control register 208 in hardware logic 212 to determine if conditions exist that preempt host [processor] 202 from resetting service processor 204." (*Id.*, Col. 4, ll. 36-40.) Examples of conditions that may preempt the resetting of the service processor 204 include when the service processor 204 is: in a special debug mode, in the process of handling a critical event, and attempting to recover from a self detected error. (*Id.*, Col. 4, ll. 40-45.) If no status exceptions (i.e. conditions that preempt the host processor

202 from resetting the service processor 204) exist, the host processor then provides a warning to the service processor 204 that a hard reset is about to commence. (*Id.*, Col. 4, ll. 46-50.) If the service processor 204 acknowledges this warning and agrees to being reset or if the service processor 204 does not respond within a timeout period, the host processor 202 resets the service processor 204. (*Id.*, Col. 4, ll. 46-65.)

In the former case, Forsman relies only on the detection, by the host processor 202, of a lack of heartbeat signals from the service processor 204 before resetting the service processor 204. In other words, Forsman fails to teach, expressly or inherently, a detection of any type of discrepancy between a current state and a current activity of a co-processor. At best, Forsman appears to teach a detection of a current activity (i.e., the lack of heartbeat signals from a service processor 204) and associates this current activity with a "not functioning correctly" condition. In the latter case, Forsman relies not on a detection of a discrepancy, but on the detection of a consistency between the lack of heartbeat signals and the lack of any preemptive conditions or status exceptions in determining that the service processor is not functioning correctly and should be reset. Because Forsman fails to teach the detection of a discrepancy between a current state of a co-processor and a current activity of the co-processor, as required by claim 1, the rejection is in error and must be withdrawn.

(ii) *The Final Office Action's Rejection*

Ignoring the plain teachings in Forman, the Final Office Action provides the following explanation of its rejection of claim 1.

> The cited portions of Forsman disclose a system in which a host processor and service processor exchange heartbeat signals (column 4, lines 9-12). The heartbeat signals indicate that a service processor is active and working correctly (column 1, lines 35-37). In other words, the heartbeat signal is an indication of the current activity of the processor. In order for an abnormality to be found, the status of the service processor must be normal (column

> 4, lines 36-45). Thus, a discrepancy exists, and the host [processor] resets the service processor and returns it to its original state and activity where it can continue to process heartbeat signals (column 4, lines 46-50). (Final Office Action, p. 9.)
>
> Furthermore, in Forsman, the absence of heartbeat signals does not automatically indicate a service processor failure, as the status register may indicate an exceptional condition .... Thus, applicant's interpretation that at the time of failure to detect heartbeat signals the current state of the processor is not active and working correctly is incorrect. Instead the current status is actually determined by status of the service processor at the moment that failure of heartbeat signal detection occurs. (*Id.*, p. 10.)

Appellant submits that the above comments constitute clear error on behalf of the Examiner because it manipulates the teachings of Forsman in an attempt to provide a *prima facie* case of anticipation against Appellant's claim 1, where none exists.

Specifically, the Examiner states that the exchange or lack of heartbeat signals is akin to Appellant's claimed current activity of a co-processor. (Final Office Action, p. 9.) Citing to column 4, lines 36-45 of Forsman, the Examiner then states that "[i]n order for an abnormality to be found, the current state of the service processor must be normal." *Id.* However, this portion of Forsman does not stand for the proposition set forth by the Examiner. Instead, this portion of the reference, as properly described above, discusses an embodiment of Forsman where the host processor 202 checks to see if there are any conditions that may preempt the host processor 202 from resetting the service processor 204.

Appellants are unable to find any teaching, express or implicit, in Forsman that stands for the proposition put forth by the Examiner. In fact, Appellants submit that Forsman appears to teach the opposite: when the host processor 202 fails to detect heartbeat signals, the current state of the service processor 204 is working improperly. (Forsman, Col. 4, ll. 13-15.) This current state is <u>consistent</u> with the current activity of the service processor 204 (i.e., not sending heartbeat signals).

At best, it appears that the Examiner is manipulating Forsman to suggest that a discrepancy of any kind is detected. As described above, Forsman is limited to: (1) resetting a service processor when a host processor fails to detect a heartbeat signal from the service processor; or (2) resetting a service processor when (a) a host processor fails to detect a heartbeat signal from the service processor and (b) the host processor determines that conditions do not exist that preempt the host processor from resetting the service processor. In the first instance, Forsman fails to detect any sort of discrepancy between two things such as a current state and a current activity of a co-processor. Instead, Forsman immediately resets the service processor. (Forsman, Col. 4, ll. 25-35.) In the second instance, Forsman resets the service processor only after determining a <u>consistent</u> condition between the lack of heartbeat signals and a lack of preempting conditions. (Forsman, Col. 4, ll. 25-65.) Each of the reasons presented immediately above demonstrate the error in the Examiner's rejection of claim 1.

B. <u>CLAIMS 28, 30, 32 AND 34 STAND OR FALL TOGETHER WITH INDEPENDENT CLAIM 26</u>

Claims 26, 28, 30, 32 and 34 stand rejected under 35 U.S.C. § 102(e) as allegedly being anticipated by Forsman. Claims 28, 30, 32 and 34 stand or fall together with independent claim 26. Claim 26 requires a circuit for monitoring and resetting a co-processor comprising:

> a hang detector module operative to detect a hang in the co-processor <u>by detecting a discrepancy between a current state of the co-processor and data in the one or more storage elements associated with the co-processor, wherein the data in the one or more storage elements represents a current activity of the co-processor</u>; and
>
> a selective processor reset module operative to selectively reset the co-processor without resetting a processor, in response to detecting a hang in the co-processor. (Emphasis added).

The Final Office Action alleges that Forsman discloses all aspects of Appellant's claim

26. (Final Office Action, pp. 2, 5.) Appellant disagrees for the same reasons set forth above. Namely, Forsman is limited to a system and method that (1) resets a service processor when a host processor fails to detect a heartbeat signal from the service processor; or (2) resets a service processor when (a) a host processor fails to detect a heartbeat signal from the service processor and (b) the host processor determines that conditions do not exist that preempt the host processor from resetting the service processor. In the first instance, Forsman fails to detect any sort of discrepancy between two things such as a current state and a current activity of a co-processor. In the second instance, Forsman resets the service processor only after determining a <u>consistent</u> condition between the lack of heartbeat signals and a lack of preempting conditions.

(i) *<u>The Final Office Action's Rejection</u>*

Once again ignoring the plain teachings in Forman, the Final Office Action provides the following explanation of its rejection of claim 26.

> Regarding claims 26, 30 and 36 … Forsman teaches that wherein the status/control register is checked after a failure to detect heartbeat signals (column 4, lines 36-45). If no status exceptions are present, then the processor is reset (column 4, lines 46-50). Examples of status exceptions include when the service processor is in a special debug mode or when the service processor is in the process of handling a critical event (column 4, lines 40-44). Both of these are indications that the service processor is not in a hang condition. Thus, checking of the status register is part of the detection of a hang in the service processor. (Final Office Action, p. 10.)

Initially, Appellant submits that the checking of the status register to determine if any conditions (e.g., status exceptions) exist that preempt the host processor 202 from resetting the service processor 204 appears to be part of detecting a hang in the service processor 202 in the Forsman embodiment where the host processor initiates the communications recovery or resetting of the service processor 204. (Forsman, Col. 4, ll. 36-50.) In the more general embodiment, described

above as the first embodiment, the checking of the status register is not performed. (*Id*., Col. 4, ll. 25-35.)

Appellant further submits that the Examiner's own rejection supports Appellant's argument. The Examiner clearly states that examples of status exceptions are indications that the service processor is not in a hang condition. (Final Office Action, p. 10.) The corollary of this statement is also true: The lack of status exceptions are indications that the service processor 204 is in a hang condition. (Forman, Col. 4, ll. 36-50.) Thus, as noted above, in the embodiment where the host processor 202 resets the service processor 204, Forsman looks for consistency between the lack of heartbeat signals and the lack of status exceptions. The lack of heartbeat signals tends to indicate a hang condition, while the lack of status exceptions confirms the hang condition. Forsman does not appear to teach the detection of a discrepancy between a current state of the co-processor and data in one or more storage elements associated with the co-processor, wherein the data in the one or more storage elements represents a current activity of the co-processor. This reason demands the withdrawal of rejection against claim 26.

C. *<u>CLAIMS 29 AND 33 STAND OR FALL TOGETHER WITH CLAIM 35</u>*

Claims 29, 33 and 35 stand rejected under 35 U.S.C. § 102(e) as allegedly being anticipated by Forsman. Claims 29 and 33 stand or fall together with claim 35. Claim 35 reads: the circuit of claim 1, wherein:

> <u>The discrepancy is detected by comparing data representing a current state of the co-processor</u> with data representing a current activity of the co-processor. (Emphasis added).

The Final Office Action alleges that Forsman discloses all aspects of Appellant's claim 1. (Final Office Action, p. 2, 7.) Appellant disagrees for the same reasons set forth above in subsections 7(A) and 7(B). Appellant further notes that Forsman appears to be silent as to comparing data representing a current state of the co-processor with data representing a current activity of the co-

processor. Instead, Forsman appears to teach recognizing the lack of a heartbeat signal as a hang or recognizing the lack of a heartbeat signal and ensuring that a hang exists by examining the contents of a status/control register for the purpose of detecting a consistency (not a discrepancy) indicative of a hang condition.

(ii) *The Final Office Action's Rejection*

In its rejection of claim 35, the Final Office Action merely points to column 4, lines 25-50. However, these are the same lines fully explained above with respect to the claims 1 and 26. As noted, Forsman does not appear to be capable of comparing data representing a current state of a co-processor with data representing a current activity of a co-processor. Instead, Forsman is limited to detecting the lack of a heartbeat signal (i.e., the lack of data) and either: (1) resetting the service processor based on the lack of a heartbeat signal or (2) resetting the service processor after confirming that the service processor is hung by not finding any status exceptions in a status/control register.

D. CLAIMS 5 AND 10 STAND OR FALL TOGETHER WITH INDEPENDENT CLAIM 11

Claims 5 and 10-11 stand rejected under 35 U.S.C. § 102(e) as allegedly being anticipated by Forsman. Claims 5 and 10 stand or fall together with independent claim 11. Claim 11 requires a circuit for monitoring and resetting a co-processor comprising, among other things:

> a hang detector module operative to detect a hang in the co-processor; [and]
>
> a halt communications module operative to halt executable instruction communications with the co-processor, in response to detecting a hang in the co-processor. (Emphasis added).

The Final Office Action alleges that Forsman discloses all aspects of Appellant's claim 11. (Final Office Action, pp. 2, 4.) Appellant disagrees because Forsman merely provides for a timeout period after determining that a hard reset of the service processor 204 is required, whereby during the timeout period, the host processor 202 waits to receive an acknowledgement that service processor 204 is ready to be reset. (Forsman, Col. 5, ll.10-23.) If no such acknowledgement is received, the host processor 202 resets the service processor 204 after the timeout period expires. Forsman appears silent as to a halt communication module operative to halt execution instruction communications with a co-processor.

(i) *The Forsman Reference*

Forsman clearly provides that after the host processor 202 determines that there are no status exceptions that preempt the host processor 202 from resetting the service processor 204, "then the host [processor] sends a signal to the service processor warning the service processor that a hard reset is about to occur." (Forsman, Col. 5, ll. 10-14.) "The host [processor] then determines whether an acknowledgement has been received or a timeout has occurred." (*Id.*, Col. 5, ll. 16-17.) "The acknowledgement indicates that the service processor has received the warning and is ready to be reset. The timeout period is a predefined interval of time that the host must wait for a response from the service processor before assuming that the service processor is not going to respond." (*Id.*, Col. 5, ll. 17-22.) "Once the acknowledgement has been received or the timeout period occurred, the host causes a hard reset of the service processor." (*Id.*, Col. 5, ll. 23-26.)

As explained, Forsman merely provides a timeout period during which the service processor 204 has an opportunity to signal to the host processor 202 that is ready to be reset.

Forsman is silent as to a halt communication module operative to halt executable instruction communications with the co-processor.

(ii) *The Final Office Action's Rejection*

Once again ignoring the plain teachings in Forman, the Final Office Action provides the following explanation of its rejection of claim 11.

> With regard to claim 11, … Applicant asserts that the disclosure of waiting a predetermined timeout period for the service processor to respond to a reset warning does not teach halting of command communications with the co-processor because communication is possible. However, the examiner maintains that this is an incorrect interpretation of the claim. In Forsman, although communication is possible, during the period that the host is waiting for acknowledgement, communication has been halted (column 5, lines 10-22 and figure 2, steps 304-308). In this period of time that the host is waiting, whether an acknowledgement is received or the process times-out, there is not other communication between the host and the service processor. Thus, all communications, including executable instruction communications are halted. (Final Office Action, p. 11.)

Appellant submits that the Examiner comments above constitute clear error because it ignores claim language. Claim 11 requires, among other things, a circuit comprising a halt communications module that is operative to halt executable instruction communications. Forsman neither teaches nor describes any such module. The Examiner improperly assumes that something like a halt communications module must exist in Forsman because communication has allegedly been halted during the timeout period. This is improper because: (1) Forsman does not state that communication has been halted; and (2) Forsman does not provide, teach or even suggest that any halt communication module exists to halt executable instruction communications with the co-processor. In fact, Forsman expressly states that communication from the service processor 202 to the host processor 204 may exist by way of an

acknowledgement message. Because Forsman does not teach what is suggested, claim 11 is in proper condition for allowance.

E. CLAIMS 12 AND 13 STAND REJECTED UNDER 35 U.S.C. § 103(A) AS ALLEGEDLY BEING UNPATENTABLE OVER FORSMAN IN VIEW OF HILL

Claims 2, 6-7, 12-13 and 24 stand or fall together with claim 1. Claims 12 and 13 are allowable for the reasons stated above with respect to claim 1.